

# ATARI COMPUTER ENTHUSIASTS

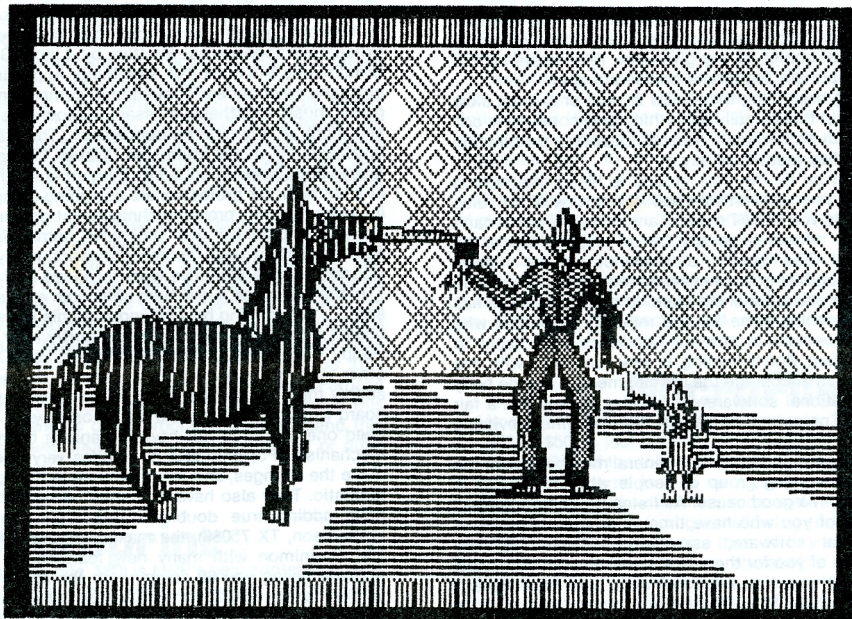
3662 Vine Maple Dr. Eugene OR 97405

---

JULY-AUGUST 1984

Editors: Mike Dunn, Jim Bumpas, Larry Gold

---



NO MEETING THIS MONTH



## ONE LAST BLAST

Two years is not a long time out of a man's life. The past two years have been very exciting and seem to have been exceptionally full of activity. My wife claims this is because I was never home (which isn't true, I distinctly remember having dinner with her last month) and I spend all of my time running around at meetings. Generally it has been a very worthwhile investment of my time.

The microcomputer industry has changed quite a lot during my two years in office. Texas Instruments sparked an industry-wide price war reminiscent of the old gas wars. This resulted in TI dropping the 99/4A computer and helped several other major computer makers lose many millions of dollars in the effort to maintain a piece of the market. Atari was not unscathed in this little fiasco; in fact Atari has been one of the more interesting companies to watch over the years. Lately I am reminded of the recipe for making steel: Take several tons of ore, add some coal and limestone, and burn the daylight out of the whole mess. The good stuff stays put and the slag floats off the top.

It seems Atari has had its share of personnel turnover and upheaval. Hopefully the new stripped down version of Atari will be "lean and mean" as advertised, and will offer another new generation of leadership in home computer design. I don't yet know what the final version of the 1450XLD will look like or act like. If it is as good as some people at Atari claim, it will be good indeed.

When I purchased my 800 it was because I had studied all of the available home and personal computers. The study convinced me the Atari had more to offer for the dollar than any other system. Over the years I have felt my decision was the correct one. The design of the Atari 800/400 is ahead of its time, and is now basically the industry standard as far as conceptual design. (Hopefully I won't get sued for saying I think the COMMODORE 64/VIC 20 and TI99/4A are basically clones of the Atari design) Graphics and sound have become a requirement among both personal and business computers within the last few years, and Atari was the first to design a sensible system for adding those capabilities to a microcomputer.

The challenge now for Atari is to once again step out in front of the crowd. The plan Atari has talked about has to do this time with the machine/human interface. The joystick, keyboard and paddle controllers are bound for retirement if Atari has anything to say about it.

The user groups also have a new challenge today. The software industry is having absolute convulsions over the amount of piracy taking place; numbers of copies of any given new release are spread around the country before any legitimate copies are sold.

The industry is trying everything they know to stop or at least slow down the rate of loss, but this is really something only consumers can do. The user groups can be heroes or villains, depending on how the group is run and how they react to piracy.

Another challenge faced by the user groups is filling the gaps in software left by the commercial software publishers. One of the rules of publishing software is you DON'T publish any software which can't be expected to sell a certain large number of copies. Since the microcomputer market has shown a tendency to be heavily entertainment oriented and fairly business oriented, there has seemed to be little room left for educational software. In particular, there has been very little programming done for the minority of people who suffer from various educational or physical handicaps. I am happy to say the new ACE President, Bob Browning, is very concerned with this issue and will be trying to encourage the development of a large body of public-domain educational software. Our library already has a fair number of educational programs, but again there are the inevitable gaps.

During my term as president, I found the general membership of the ACE is a dedicated and giving group of people who are willing to devote time and energy in a good cause. As I step down from office I want to encourage all of you who have time to program to work at developing educational software, especially software for the handicapped. I thank all of you for the support over the past two years, and for being patient with me. You will continue to hear from me, as I am now a member of MICROBITS Peripheral Products, a corporation with a lot of exciting developments in the offing. Until next time, Thank you all.

—Kirt E. Stockwell  
Past President

## News and Reviews

by Mike Dunn, Co-Editor

Almost every day I receive a letter begging for information on programs in our exchange library or sold by others which are helpful for Learning Disabled or Handicapped children or adults. These are from both parents and teachers. We have many educational programs and games, but I do not know which ones might be helpful. There are also commercial and "home-brew" devices which might be helpful.

**We want to devote the October issue to the Learning Disabled and Handicapped.** Please send us your experiences, articles, circuit diagrams, reviews of our software and commercial software, etc. Any items you think might be helpful to others with these problems will be appreciated by all. I will also ask Stan Ockers, Sydney Brown, Ruth Ellsworth, Dale Lutz and others if they can come up with some suitable programs for this very important special issue.

At the last ACE meeting, we held elections for President, and Robert Browning is now our new leader. Bob has been very active in E.R.A.C.E., and will lead our group to new heights. Also seen at the meeting were demonstrations of the new Atari/Synapse programs, **Syn-Calc** and **Syn-File**. **Syn-Calc** is a very easy to use, powerful spreadsheet and **Syn-File** is a database system replacing the popular **File-Manager 800**. Both are written in FORTH, have built in tutorials leading you through the various functions, have many built-in help features, etc. They look very nice, cost \$99 each. I have not worked with them; the Demo was by a local computer store. We will review them extensively if we get a review copy; if not, maybe one of you will like to do so. Also seen at the meeting was the fabulous new **MindSet** computer! Demonstrated was a demo version of a fantastic game from Synapse also called **Viper!**. Featuring 3-D graphics, extremely smooth scrolling, it had almost a motion-picture effect rather than a computer generated game; very impressive. Also a CAD-type drawing program by **MindSet** and **Time Arts**, **Lumena**, is a professional color graphics package costing \$400. This turns the **MindSet** into a \$50,000-type CAD computer. The demo pictures with the programs look like color slides — very beautiful! Local dealers are Computer Palace 503-683-5361 and ACE Computer 503-343-5191, and they are in stock.

We received some more very nice books from Prentice-Hall, Englewood Cliffs, N.J. 07632. For general use, **The Encyclopedia of Microcomputer Terminology**, a sourcebook for business and professional people has more than 4000 terms defined and has exhaustive appendices. Written by Linda Gail Christie and John Cristie, this \$10 book has everything. Appendices include a bar-code glossary, color codes for electronic components, logic functions and gates, music synthesis glossary, videodisk technology and glossary, and more. A very nice reference book. The other is a beautiful book, invaluable to those interested, **Visual Display Terminals: Usability issues and concerns** edited by Bennett et al., this \$28 hard cover book is a textbook for professionals designing computer display setups. It includes specific, proven techniques introducing VDT's, including the social and organizational problems, human factors, etc. Has much data supporting their recommendations, many charts, formulas, etc. Appears to be a definitive work for those in the field.

In the last issue, Steve Berg sent in an article ("Give your old Atari a shot...") describing how he upgraded his Atari and drive with surplus boards from American TV Sales. Shortly after, I received an ad from Happy Computing (POB 1268, Morgan Hill, CA 95037) with, among other very interesting items, an **Analog upgrade disk drive Kit**. They supply the upgrade boards for a new powersupply and speed control board to vastly improve the operation of your drive. You can tell you need one if you do not have the analog board above the disk drive mechanism. For only \$50, it includes very fine directions on how to make the changes, and the difference in the operation of the board is fantastic. They also have a "Happy" enhancement for the Atari 1050 drive adding true double-density. CDY Consulting (421 Hanbee, Richardson, TX 75080), the makers of **OMNIMON!**, now have a new 16K Ominimon with many new features, as well as a 80 column upgrade giving you 80 columns on the screen. There is even a new version for the 600XL and 800XL due out soon. I have not seen these new products yet, but if they are as good as the old one, they should be fantastic.

The more I use the new **Letter Perfect Version 6** by LJK (see June issue for review), the more I am impressed. Their new **DataPerfect** is also a peach, with all the bugs removed and many new features — still under review.

Remember, the next issue of ACE will be in September. Have a nice summer.



## BUMPAS REVIEWS

**TOP DOS** (Eclipse Software, 1058 Marigold Court, Sunnyvale, CA 94086, \$50) is an enhanced Disk Operating System for the Atari. If you're familiar with Atari DOS 2.0, you will be able to use this one immediately as it presents a menu format which is very similar and contains the familiar commands. The documentation tells you TOP DOS occupies the same space in memory as Atari DOS, so all Atari software should be fully compatible. TOP DOS is not compatible with the very early Ataris which still use the Revision A OS ROM. TOP DOS will tell you upon booting when you have Rev. A.

One of the first differences you will probably notice is the entire screen scrolls when you begin entering commands. Atari DOS only scrolls four lines at the bottom. For example, you may not have to keep calling up a disk directory to see the next file upon which you want to work. You may limit commands to one line as you enter them, keeping more of the history on the screen.

Control of directory format is one of the more interesting powers of TOP DOS. Directories may be displayed in up to 6 columns. Two columns is the default. Four columns may be used in an 80-column display. Six columns may be printed out. Directories may be alphabetized on disks formatted in TOP DOS format. Unfortunately, the TOP DOS format prevents use of the disk by any program which uses Atari DOS. But TOP DOS can also format a disk in the Atari format, and TOP DOS can use disks formatted with Atari DOS.

Other added commands permit the user to create command files which execute a series of operations (formatting, copying files, etc.) with one command. Bytes in memory may be examined and changed directly from the DOS menu. Deleted files may be restored with the Undelete command.

The TOP DOS system disk contains extensive help files. The "?" key pressed after calling up any command will access a help file describing the function. Pressing "T" after getting any error message will print out a short description of the error (saving the effort of having to look up the error).

The command menu also displays additional information. Each active disk drive is listed, with the default drive indicated. Each drive shows single, double or quad density. The number of buffers available for open files is shown, as are MEMLO and MEMTOP. Various other statuses which may be toggled on and off are shown: MEM.SAV, Auto-RS232, Verify Write, DOS Resident, and Bypass Cartridge. Control over each of these functions indicate a very powerful DOS. This review has insufficient space to describe them further.

I am pleased to say the documentation for this DOS seems excellent. In a manual of more than 80 digest sized pages are complete descriptions of the use of all the functions. And examples of use are included to help make these descriptions clear. I am very impressed by the documentation. The only improvement I can suggest is an index.

TOP DOS does not seem to adjust automatically to disks of different densities. The Indus drive I use switches density automatically when a disk is inserted. But to access the disk, you must reboot, or else call up the Status menu and toggle the status for that drive. We tested several programs using TOP DOS, on an XL as well as a regular 800. We encountered one problem on both machines when running Telengard by Avalon Hill. The text display disintegrates when using TOP DOS, and does not disintegrate with Atari DOS. This indicates some problem in handling graphics in some way. Perhaps the excellent programmers at Eclipse can find and correct this problem, which might extend to other programs. I recommend this DOS for advanced users, and those who are using double or quad density disk drives.

## Microprint

Microprint (also from Microbits, \$80) is MPP's new printer interface. When you first see it you think it is a printer cable with a large plug at one end. Inside the large plug is housed all the electronics needed to make the interface work.

I tried it on many types of printers: dot matrix, ink jet and daisy wheel. The only thing all of the printers had in common was they were all parallel with centronics type plugs.

If one only needs an interface to do one job and is not fancy, but does the job well and at a reasonable price then I highly recommend this interface to them.

—Larry Gold

## SYNCALC

Well, ATARI really did something right! They are marketing the new worksheet program by Synapse. **SynCalc** (\$100) does everything possible with Visicalc, and a whole lot more. Many of the features of Lotus 1-2-3 are incorporated into SynCalc. For instance, you can alter column widths on an individual column basis, rather than for the entire worksheet. You can also type descriptive labels without having to stop and jump the cursor as in Visicalc. Anyone who has had to cope with this Visicalc limitation will probably buy the program for this reason alone.

Other features of SynCalc include many more built-in formulas, sorting capability, printing with or without column and row headings, and others too numerous to mention here.

For those of you who hate to memorize command instructions, the program is completely menu driven. You can execute commands simply by moving the cursor. If you are an expert user, you can bypass the menu and use the commands directly (they are very similar to Visicalc and Lotus).

SynCalc can also share data with SynFile, SynTrend and AtariWriter. I have not had a chance to work with this feature.

You get all the above, plus one additional bonus if you have Visicalc files; yes, you can copy Visicalc files to SynCalc files, and do away with a lot of re-entering. The copied files will require some modification of formulas to make them run on SynCalc.

The program comes with an excellent instruction manual which consists of three progressively more difficult tutorials. The first tutorial is short enough so you can have your first worksheet ready in minutes.

SynCalc is a powerful, user-friendly spreadsheet program which has many of the capabilities of Lotus 1-2-3.

— Jim Landen

## 7800 PROSYSTEM

The **7800 PROSYSTEM** is a new game machine from Atari (\$150) which will run all 2600 games without adaptor, and has many new features. It has a proprietary "Maria" chip which permits more moving objects per scan line — up to 100 independently moving objects on the screen at one time. They advertise more realistic color and more programming flexibility than ever before on any videogame or home computer system.

A keyboard will be available with 4k RAM, expandable to 20k. They plan word processing, creative learning, and personal development software for this system. Most XL line peripherals and accessories will also be compatible with the 7800. It comes with a serial I/O expansion port to allow it to be upgraded with future game technologies.

Video game machines are still selling — at the right price. A friend of mine just bought two 2600s for about \$25 (total!). But I don't know if people will pay a home-computer price for a video game machine which can (for extra cost) be made into a 20k computer. I haven't seen it yet, but if the graphics and action are really a great leap beyond what is available now, then the machine will probably be successful. When you start using this machine, let us all know what you think of it.

— Jim Bumpas, Co-Editor

## MicroFiler

MicroFiler (Microbits Peripheral Products, 225 W. Third St., Albany, Or. 97321, \$50) is just what its name implies, a filer program or small data base. One of the things making this program different from other programs of its type is it is on a cartridge, and because it is on a cartridge it will work with either a cassette recorder or disk drive.

Here is a menu driven program allowing you to browse through files, change the fields to customize them to fit your needs, prints labels or lists. It allows you to sort or alphabetize files, does addition or subtraction and even averages numeric information. Permanent storage can be made to either disk or cassette. A lot from one ROM cartridge.

The manual covers everything step by step so you learn how to use this program with a minimum of fuss and bother. For all those who only have a cassette recorder this is an ideal program for you as you only need use your cassette to store your data on, and a person with a 400 and 16K can use this program the same as anyone else with larger systems. This program is for anyone who needs a small data base for lists, expense accounts, collections, recipes or anything you need for storing data.

—Larry Gold



## Lights, Camera, ACTION!

Recently I jumped aboard the ACTION! bandwagon and purchased the language. I haven't had much time to do any large program development with it, but already I have fallen in love with it. With the Programmer's Aid disk and the runtime package available, I can't think of anything to improve. This language is really where it's at — if you have grown out of BASIC, get ACTION!

To give you a brief sampling of ACTION!, I have submitted two simple programs. The first converts TinyText files to TextWizard/AtariWriter format. It illustrates the use of the OPEN, PUT and GET routines in ACTION!. Note the smooth flow of the ACTION! code in this one. When typing it in, note that in PROC allout the things which look like ones are actually Ls. (Experienced ACTION! programmers will probably laugh at this effort of mine — when I wrote it I didn't know the shortcuts of the language, and it still runs fast anyway, so I didn't change the code). All the program does is take out the carriage returns Tiny Text puts in, and change the Tiny Text Control-S and Control-E to returns. The dots still have to be removed manually, and any format controls changed appropriately.

The second program lists ACTION! source files to single sheets, and numbers the pages. I used it to generate listings for inclusion in a hand-in school physics project — I couldn't find any other way to list the source files to single sheets. In fact, this program could be used to dump any pre-formatted text file to printer.

In conclusion, I plan to use ACTION! as my sole language from now on, except for very small 'throw away' programs where BASIC is better. But for anything very large, swing into ACTION!.

—Dale Lutz  
Canada

## Double Bounce

Double Bounce is a game for two players. The object for each is to bounce on top of a floating cloud and climb down the ladder on the left of the screen. Ten points are gained each time you do this. Choose some reasonable point goal before you start.

Joysticks in ports 1 & 2 control trampolines which bounce the players. Pushing up when contact is made causes the player to go higher — down decreases the height. You needn't push up or down to maintain a constant height. The fire button releases the player for the first bounce. Miss a trampoline and you fall in the quicksand below. You will then have to wait for the other player to suffer the same fate. When this happens, two new bouncers will appear on screen.

The difficulty for each player can be selected at any time. Pushing 'OPTION' changes the difficulty for the player on the left, while 'SELECT' changes that of the right hand player. Higher difficulty means the players move horizontally faster. The 'START' key will restart the game. Oh yes, beware of the big green bird!

— Stan Ockers

## BULL ANTS

In the Bull Ants game this month, you and your friend are ants and you have lived happily in your nest for a long time, but now the construction workers have moved into the neighborhood and are using explosives to dig trenches for the foundations of a new shopping center. Another problem are the bull ants which have moved into your next because their home was destroyed.

Use joysticks 1 and 2 to control the red and green ants. You must move your five eggs from the bottom of the nest to safety at the top of the screen. On the way up you must watch out for cave-ins and the bull ants.

If you are hit by a cave-in, you will be sent to the top of the screen. If you are stung by a bull ant, you will be paralyzed and will have to wait until the next explosion to shock you out of it.

If you were carrying an egg in either case, you will lose it and it will be represented by a black egg appearing at the top of the screen on your side.

The winner is the one who gets the most eggs to the top safely. If both players get the same number, the winner will be the first to get them there.

— Sydney Brown

## CASSETTE MENU

(Reprint: June, 1984 Santa Maria/Lompoc ACE)

Some months back I played around with the idea of a Cassette Operating System to pick up where I believe Atari left off. Options such as controlling the baud rate, increasing sector density, allowing direct copying from disk to cassette, the ability to specify which file to run were just some of the things the Atari 410/1010 program recorder is capable of when paired with the right software. After I got my disk drives, I really started thinking about how to develop a COS patterned loosely on DOS. When I found how difficult it is to put everything into page six of memory and ran into other numerous walls, I put it aside thinking I could try again later.

While doing some of the functions mentioned earlier requires machine code, one function can be done in Basic as a utility. The Atari system can be configured to run a two-stage program which displays a menu of the programs on a cassette tape and allows a search of the tape to load and run a specified program. To many cassette owners this means the end of trying to position a tape properly to get a program to load after losing the program's place on the tape because of a faulty tape counter.

This program is in two parts. It is based on the menu program I devised for my own use with disk-based programs. The original program displayed the disk menu and loads and runs a program based on which value is given through the keyboard. The first part of this program will display the menu of the programs you have on a cassette tape. This part allows you to easily update the menu itself and will always occupy the same amount of space on the tape. This way you will not have to re-record all the programs or worry about how close together you should put your programs on your tape. The second part of the program is the search utility which will seek out the program you indicate. It will then load and run the program without further help from the user.

In order to allow this program to work, two conditions are necessary. First, the programs must be saved to cassette using SAVE "C:". This allows the second condition to be met, the use of a string function based on a REM statement. The REM statement is compared to the input provided by the user in response to the menu program. If a match is made, the desired program is loaded in and executed. If no match occurs, the computer continues reading the tape until the desired program is located.

Type in Listing 1 just as it appears, remembering to type the "A" through "Z" letters inside the menu as inverse characters. When you finish typing in the program, save it with SAVE "C:", not a CSAVE! Do not advance the tape at this point. Type NEW to clear memory and type in Listing 2. Save with a SAVE "C:" again. The utility is now ready for use.

To update the menu portion in the first part just issue a LOAD "C:" command, make the desired changes and save the menu display program back to tape with a SAVE "C:". I suggest moving the tape manually until the leader/tape junction appears (usually a white square) and lining it up with the pad located in the center of the tape path housing to make sure you save it back to the same place on the tape.

To allow the utility to seek out the program you want to use requires the use of a REM statement in a manner which is different from what you are probably used to. When you have typed in or loaded the program you want to use with the utility later, do the following:

The first line of the program must be in the form: "1 REM (filename)". Type in: LIST "C:",1,1 and press Return. The first line containing the REM string will be saved to the cassette recorder. It will serve as the program identifier. Now save the entire program with a SAVE "C:" command. This will allow the autorun feature already discussed. Update the menu portion as already mentioned.

Now when you want to run the program just type in RUN "C:" after positioning the tape at the beginning. The menu will appear with the program names. Press Return when you are ready and type in the name of the program at the prompt. When the Atari has found your program, it will issue a number of beeps from the console speaker, load the program and run it for you. If the program you want is a binary file, the display will stop and display the instructions for you to load it. If you do nothing, the program will assume it is a normal program and continue on its merry way while you sit back and relax.

Two last notes. You will notice a lack of beeping sounds while the program is doing its searching. This is done to give you some peace and quiet while it does its work. After all, who wants to listen to I/O noise for as long as it takes to load some programs when it's not really necessary? Finally, you do not need to leave a lot of tape between your programs, they can be placed end to end. Just note where you left off the last time you saved a program and save the next program just after that point. Now sit back and let your Atari do the work for you.

— Jay Torres



by Sydney Brown

```

0 ? "R":SP=1:GOSUB 2000 ,4
200 ST=STICK(0):POKE 53278,0:A1=A1+SP; 299 POKE 77,0:RN=INT(150*RND(0)):GOTO
IF A1>215 THEN A1=10 200
201 IF GOT1=1 THEN 209 300 IF E1=1 THEN 320
202 POKE 53250,A1:IF ST=14 AND V1<0 TH 302 LOCATE H1,V1+1,Z:IF Z<(122 OR H1)>1
EN LOCATE H1,V1-1,Z:IF Z=32 THEN COLOR 4 THEN RETURN
32:PLOT H1,V1:V1=V1-1:GOTO 209 305 GOSUB 350:E1=1:M1=M1+1:COLOR 32:PL
203 IF ST=13 AND V1<22 THEN LOCATE H1, 0T H1,V1+1:IF M1=5 AND FF=0 THEN FF=1
V1+1,Z:IF Z=32 THEN COLOR 32:PLOT H1,V 309 RETURN
1:V1=V1+1:GOTO 209 310 IF E2=1 THEN 330
205 IF ST=11 AND H1<0 THEN LOCATE H1-1 312 LOCATE H2,V2+1,Z:IF Z<(122 OR H2)<1
,V1,Z:IF Z=32 THEN COLOR 32:PLOT H1,V1 4 THEN RETURN
:H1=H1-1:GOTO 209 315 GOSUB 360:E2=1:N2=N2+1:COLOR 32:PL
207 IF ST=7 AND H1<19 THEN LOCATE H1+1 0T H2,V2+1:IF N2=5 AND FF=0 THEN FF=2
,V1,Z:IF Z=32 THEN COLOR 32:PLOT H1,V1 319 RETURN
:H1=H1+1:GOTO 209 320 IF V1>2 THEN RETURN
209 COLOR 247:PLOT H1,V1:IF V1=2 OR V1 325 GOSUB 350:COLOR 250:PLOT M1-1,0:E1
=22 THEN GOSUB 300 325 GOSUB 350:COLOR 250:PLOT M1-1,0:E1
210 ST=STICK(1):A2=A2-SP:IF A2<10 THEN :GOTO 600
A2=200 329 RETURN
211 IF GOT2=1 THEN 219 330 IF V2>2 THEN RETURN
212 POKE 53251,A2:IF ST=14 AND V2<0 TH 335 GOSUB 360:COLOR 218:PLOT 20-M2,0:E
EN LOCATE H2,V2-1,Z:IF Z=32 THEN COLOR 20:52=52+1:EG2=EG2+1:IF 552=5 THEN P0
32:PLOT H2,V2:V2=V2-1:GOTO 219 P :GOTO 600
213 IF ST=13 AND V2<22 THEN LOCATE H2, 339 RETURN
V2+1,Z:IF Z=32 THEN COLOR 32:PLOT H2,V 350 FOR M=15 TO 0 STEP -1:SOUND 0,14,1
2:V2=V2+1:GOTO 219 0,M:NEXT M:SOUND 0,255,0,4:RETURN
215 IF ST=11 AND H2<0 THEN LOCATE H2-1 360 FOR M=15 TO 0 STEP -1:SOUND 0,10,1
,V2,Z:IF Z=32 THEN COLOR 32:PLOT H2,V2 0,M:NEXT M:SOUND 0,255,0,4:RETURN
:H2=H2-1:GOTO 219 400 FOR MZ=250 TO 5 STEP -7:SOUND 0,MZ
217 IF ST=7 AND H2<19 THEN LOCATE H2+1 ,2,14:NEXT MZ:SOUND 0,255,0,6:IF Z=216
,V2,Z:IF Z=32 THEN COLOR 32:PLOT H2,V2 THEN 450
:H2=H2+1:GOTO 219 405 H1=9:V1=2:COLOR 247:PLOT H1,V1
219 COLOR 216:PLOT H2,V2:IF V2=2 OR V2 410 IF E1=0 THEN RETURN
=22 THEN GOSUB 310 415 E1=0:COLOR 122:PLOT M1-1,0:EG1=EG1
240 A1=A1+SP:IF A1<215 THEN A1=10 +1
242 A2=A2-SP:IF A2<10 THEN A2=200 449 RETURN
244 POKE 53250,A1:POKE 53251,A2 450 H2=10:V2=2:COLOR 216:PLOT H2,V2
250 IF PEEK(53254)<>0 OR PEEK(53255)<> 460 IF E2=0 THEN RETURN
0 THEN GOSUB 500 465 E2=0:COLOR 122:PLOT 20-M2,0:EG2=EG
276 IF EG1=5 AND EG2=5 THEN 600 2+1
278 IF PEEK(53279)=6 THEN 699 499 RETURN
279 IF PEEK(53279)=5 THEN GOSUB 10000: 500 IF (PEEK(53254)=4 OR PEEK(53255)=4
POSITION 9,0:? #6;"S";SP; ) AND GOT2=0 THEN GOSUB 460:GOT2=1:GOT
280 RN=RN+1:IF RN<160 THEN 200 0 510
281 SOUND 0,49,8,15:FOR M=1 TO 21:POKE 502 IF (PEEK(53254)=8 OR PEEK(53255)=8
DL,120:FOR MM=1 TO 2:NEXT MM:POKE DL, ) AND GOT1=0 THEN GOSUB 410:GOT1=1:GOT
282 FOR MM=1 TO 2:NEXT MM:NEXT M 0 510
282 FOR M=0 TO 5:COLOR 32:PLOT XY(0,M) 504 IF (PEEK(53254)=12 OR PEEK(53255)=
,XY(1,M):XY(0,M)=INT(20*RND(0)):LOCATE 12) AND (GOT1=0 OR GOT2=0) THEN GOSUB
XY(0,M),XY(1,M),Z 410:GOSUB 460:GOT1=1:GOT2=1:GOTO 510
283 SOUND 0,49,8,15-MM*3:FOR MM=1 TO 10 509 GOTO 549
:NEXT MM:COLOR 95:PLOT XY(0,M),XY(1,M) 510 FOR M=30 TO 0 STEP -1:SOUND 0,7,6,
:IF Z<32 THEN GOSUB 400 M/2:NEXT M:SOUND 0,255,0,4
285 NEXT M:GOT1=0:GOT2=0:SOUND 0,255,0 549 POKE 53278,0:RETURN

```



BULL ANT con't.

```

      ^ ^ ^ ^ ^"? #6;" ^ ^ ^ ^
^":POKE 711,92:POKE 709,204
3005 POKE 756,CB:POSITION 4,21:? #6;"5
ETTING UP"
3101 FOR B=0 TO 511:IF B>439 THEN READ
D:POKE A+B,D:NEXT B:GOTO 3110
3106 POKE A+B,PEEK(57344+B):NEXT B
3110 POKE 756,CB
3150 DL=PEEK(560)+256*PEEK(561):POKE D
L,240:FOR W=DL+7 TO DL+28 STEP 3:POKE
W,134:NEXT W:ZB=CB*256
3155 FOR W=53248 TO 53255:POKE W,0:NEX
T W:FOR W=ZB+1536 TO ZB+2048:POKE W,0:
NEXT W:POKE 53277,3:POKE 559,62
3160 POKE 706,12:POKE 707,12:POKE 5325
8,0:POKE 53259,0
3162 RESTORE 1000:FOR W=1 TO 8:READ D:
POKE ZB+1599+W,D:POKE ZB+1647+W,D:NEXT
W
3163 FOR W=1 TO 8:READ D:POKE ZB+1855+
W,D:POKE ZB+1951+W,D:NEXT W
3170 POSITION 1,20:? #6;"select---SPEE
D = ";SP:FOR W=1 TO 35:NEXT W
3171 POSITION 1,21:? #6;"START---START
GAME"
3175 IF PEEK(53279)=5 THEN GOSUB 10000
:GOTO 3170
3189 IF PEEK(53279)<6 THEN 3175
3191 POSITION 0,0:? #6;"K ^ ^ ^]
      ^ ^ ^] ^ ^ ^] ^ ^ ^] ^ ^ ^]
      ^] ^ ^ ^] ^ ^ ^] ^ ^ ^] ^ ^ ^]
3192 FOR W=5 TO 19 STEP 2:POSITION 0,W
:? #6;"AAAAAAAAAAAAAAAAAAAAA":COLOR 32:
PLOT INT(RND(0)*10),W
3193 PLOT INT(RND(0)*10)+10,W:NEXT W:P
OSITION 0,21:? #6;"AAAAAAAAA AAAAAAAA
^":? #6;"ZZZZZ^ ace ^ZZZZZ";
3195 POKE 54286,192:H1=0:V1=22:H2=19:V
2=22:COLOR 247:PLOT H1,V1:COLOR 216:PL
OT H2,V2:A1=0:A2=200:EG1=0:EG2=0
3199 RM=INT(150*RND(0)):E1=0:E2=0:N1=0
:N2=0:SOUND 0,255,0,4:FF=0:S1=0:S2=0:R
ETURN
4000 RETURN
10000 SP=SP+1:IF SP>4 THEN SP=1
10001 IF PEEK(53279)=7 THEN RETURN
10002 GOTO 10001

```

LISTER by Greg Menke

```

; **** LISTER ****
; by Dale Lutz

; This program will list an ACTION!
; source file to single sheets
; on an Epson Printer

```

BYTE a,line,page,b

BYTE ARRAY str(240),  
title(40)

```

PROC inputname()
Graphics(0)
PrintE("Please enter the filename of the ")
PrintE("ACTION! source file (in the form ")
PrintE("D:filename.ext)")
PrintE(" ")
Print(" -->")
Input$(str)
PrintE(" ")
PrintE("Insert SOURCE disk, press RETURN")
a=Input$(a)
Close(2)
Open(2,str,4,0)
RETURN

```

```

PROC wait_for_sheet()
PrintE(" ")
PrintE("Insert SHEET, press RETURN")
Print(" -->")
a=Input$(a)
RETURN

```

```

PROC margin()
FOR b=1 to 8
DO
PutD(1,32)
OD
RETURN

```

```

PROC center (BYTE ARRAY cen)
page==+1
margin()
PutD(1,27)
PrintD(1,"-a") ;set underline on
PrintD(1,cen)
PrintD(1," Page ")
PrintD(1,page)
PutD(1,27)
PrintD(1,"-") ;set underline off
PutD(1,0)
PrintDE(1," ")
PrintDE(1," ")
RETURN

```



```

PROC print_sheet()
  center(title)
  FOR a=1 TO 51
    DO
      InputSD(2,str)
      margin()
      PrintDE(1,str)
    OD
  RETURN

PROC inputtitle()
  PrintE("Please enter the title to be")
  PrintE("printed on the top of each page ")
  PrintE(" ")
  Print("  -->")
  InputS(title)
  RETURN

PROC main()
  DO
    page=0
    inputname()
    inputtitle()
    PrintE(" ")
    PrintE("Ready printer, press RETURN")
    Print("  -->")
    a=InputB()
    PrintE(" ")
    Close(1)
    Open(1,"P:",8,0)
    PutD(1,27)
    PrintD(1,"8") ;shut off paper out
    WHILE EOF(2)=0
      DO
        wait_for_sheet()
        print_sheet()
      OD
      PrintE(" ")
      Print("Press RETURN for another listing.")
      a=InputB()
      PrintE(" ")
    OD
  RETURN

```

NO MEETING THIS MONTH

```

;This program converts a TinyText
;file to Text Wizard format

```

```

;some global variables

```

```

BYTE a,b,c

```

```

BYTE ARRAY str(130)

```

```

CARD mem,count,1

```

```

PROC linein()
  InputSD(1,str)
  RETURN

```

```

PROC out()
  FOR a=1 to str(0)
    DO
      Poke(mem+count,str(a))
      count==+1
    OD
  RETURN

```

```

PROC allout()
  mem=$4000
  FOR I=mem to mem+count
    DO
      a=Peek(1)
      PutD(2,a)
    OD
  RETURN

```

```

PROC inputname()
  PrintE("Please enter the filename of the ")
  PrintE("original TinyText file (in the form ")
  PrintE("D:filename.ext)")
  PrintE(" ")
  Print("  -->")
  InputS(str)
  PrintE(" ")
  PrintE("Insert SOURCE disk, press RETURN")
  a=InputB()
  Close(1)
  Open(1,str,4,0)
  RETURN

```

```

PROC outputname()
  PrintE("Please enter the filename under which")
  PrintE("to save the converted file (it will ")
  PrintE("be in Text Wizard format).")
  PrintE(" ")
  Print("  -->")
  InputS(str)
  PrintE(" ")
  PrintE("Insert DESTINATION disk, press RETURN")

```



# STAN OCKERS DOUBLE BOUNCE

```
; DOUBLE BOUNCE
; Stan Ockers 6-84
; Written in Action (c) 1983 ACS
; ACE Newsletter, 3662 Vine Maple Dr.
; Eugene, OR 97405 July 84 $12 year
;
MODULE
BYTE accnt0,accl0,rtlim,ltlim,toplim,
accnt1,accl1,hpos1m,j,k,audf1=$D200,
botlim,pmpage,vcount=$D400,hdy0,dif0,
hsl00,hpos0m,byt,tflg,hdy1,dif1,
m0p1=$D000,m1p1=$D007,m2p1=$D00A,flip,
m3p1=$D00B,hitclr=$D01E,hitflg0,hsl01,
p2p1=$D00E,p3p1=$D00F,stk0,stk1,flip0,
status0=[0],status1=[0],scnt0,scnt1,
hpos0=$D000,hpos1=$D001,hpos2=$D002
hpos3=$D003,hposm0=$D004,hpos0,hpos1,
hposm1=$D005,hposm2=$D006,hpos2,hpos3,
hposm3=$D007,hitflg1,hitflg2,clk=20,
audf2=$D202,aflg0=[0],aflg1=[0],auf0,
audc1=$D201,audc2=$D203,auf1,auc0,
consol=$D01F,auc1,flip1,ficnt0,ficnt1,
vpos3
INT vdir0,vpos0,delv0,hdir0,delh0,
vdir1,vpos1,delv1,hdir1,delh1,birdir,
vdir3
CARD source,dest,pmbase,cnt,score0,
score1
BYTE ARRAY manbou=[0 20 42 20 0 20 42
0 20 20 54],walk1=[24 20 25 62 00
152 24 20 36 36 54],carry=[12 31 125
235 138 136 0 0 0 0],falling=[0 20
42 20 73 62 0 20 54 65 0],man0([1]),
tramp=[$F0 $F0 $F0 0 0 $0F $0F $0F],
bird1=[0 196 107 60 122 16 40 0],
cloud=[16 56 100 214 191 126 60],
climb1=[0 20 62 28 72 62 9 28 20 52
61,climb2=[0 20 62 28 9 62 72 28 20
22 40],walk2=[24 20 26 60 00 24 24
120 72 72 12],bird2=[0 35 214 60 94
0 20 0],man1([1])

PROC Setdif0() ; difficulty plyr 0
dif0==+1 IF dif0=4 THEN dif0=1 FI
hdy0=5-dif0 Poke(656,2) Poke(657,2)
Print("DIF=") PrintB(dif0)
RETURN

PROC Setdif1() ; difficulty plyr 1
dif1==+1 IF dif1=4 THEN dif1=1 FI
hdy1=5-dif1 Poke(656,2) Poke(657,33)
Print("DIF=") PrintB(dif1)
RETURN

PROC Begin0() ; restart plyr 0
vpos0=100 hdir0=0 delv0=0 hpos0=50
delh0=1 vdir0=1 accnt0=1 hitclr=0
hitflg0=0 scnt0=9 status0=0 ficnt0=9
SetBlock(pmbase+1024,256,0)
RETURN

PROC Begin1() ; restart plyr 1
vpos1=175 hdir1=0 delv1=0 hpos1=50
delh1=1 vdir1=1 accnt1=1 hitclr=0
hitflg1=0 scnt1=9 status1=0 ficnt1=9
SetBlock(pmbase+1200,256,0)
RETURN

PROC Init() ; restart everything
accl0=10 rtlim=205 ltlim=45
toplim=5 botlim=205 Begin0()
accl1=10 hpos0m=120 Begin1()
hpos0=50 vdir0=1 hdir0=0 delv0=0
hitflg0=0 hitflg1=0 score0=0 score1=0
hpos2=175 hpos3=175 hpos1m=160
dif0=0 Setdif0() dif1=0 Setdif1()
flip0=0 flip1=0 birdir=1 vpos3=100
vdir3=1
RETURN

PROC Pminit() ; Player Missiles
pmbase=pmpage*256
Poke(54279,pmpage) Poke(559,62)
Poke(53277,3) Poke(623,17)
SetBlock(pmbase+968,1280,0) vpos3=100
dest=pmbase+964 source=tramp
MoveBlock(dest,source,0) Poke(711,33)
dest=pmbase+1620 source=cloud
MoveBlock(dest,source,7)
Poke(704,50) Poke($D00C,$FF)
Poke(705,46) Poke(706,13)
Poke($D00A,1) Poke($D00B,1)
Poke(707,192)
RETURN

PROC Parab0() ; bounce plyr 0
vpos0==+delv0*vdir0 audf1=100+vpos0/2
IF vpos0 MOD 4=0 THEN audc1==+1
IF audc1<$A0 THEN audc1=$A0 FI FI
IF vpos0<toplim THEN vpos0=toplim FI
IF vpos0>botlim THEN vpos0=botlim
status0=5 FI
accnt0==+1
IF accnt0=0 THEN accnt0=accl0
IF vdir0<0 THEN delv0==+1
IF delv0<0 THEN delv0=0 vdir0=1
hitclr=0 hitflg0=0 FI
ELSE delv0==+1 FI FI
hsl00==+1 IF hsl00=0 THEN hsl00=hdy0
hpos1==+delh0*hdir1
IF hpos1>rtlim THEN hpos1=rtlim
hdir1=-1 FI
IF hpos1<ltlim THEN hpos1=ltlim
hdir1=1 FI FI
IF hitflg1=0 THEN
IF (m2p1&2)=2 THEN hdir1=1
hitflg1=1 vdir1=-1 FI
IF (m3p1&2)=2 THEN hdir1=-1
hitflg1=1 vdir1=-1 FI
IF hitflg1=1 THEN audc2=$AF
IF (stk1&1)=0 AND delv1<6 THEN
delv1==+1 FI
IF (stk1&2)=0 AND delv1>1 THEN
delv1==+1 FI FI FI
IF (p2p1&2)=2 THEN status1=3
score1==+10 Position(14,1)
PrintC0(6,score1)
PROC Parab1() ; bounce player 1
vpos1==+delv1*vdir1 audf2=100+vpos1/2
IF vpos1 MOD 4=0 THEN audc2==+1
IF audc2<$A0 THEN audc2=$A0 FI FI
IF vpos1<toplim THEN vpos1=toplim FI
IF vpos1>botlim THEN vpos1=botlim
status1=5 FI
accnt1==+1
IF accnt1=0 THEN accnt1=accl1
IF vdir1<0 THEN delv1==+1
IF delv1<0 THEN delv1=0 vdir1=1
hitclr=0 hitflg1=0 FI
ELSE delv1==+1 FI FI
hsl01==+1 IF hsl01=0 THEN hsl01=hdy1
hpos1==+delh0*hdir1
IF hpos1>rtlim THEN hpos1=rtlim
hdir1=-1 FI
IF hpos1<ltlim THEN hpos1=ltlim
hdir1=1 FI FI
IF hitflg1=0 THEN
IF (m2p1&2)=2 THEN hdir1=1
hitflg1=1 vdir1=-1 FI
IF (m3p1&2)=2 THEN hdir1=-1
hitflg1=1 vdir1=-1 FI
IF hitflg1=1 THEN audc2=$AF
IF (stk1&1)=0 AND delv1<6 THEN
delv1==+1 FI
IF (stk1&2)=0 AND delv1>1 THEN
delv1==+1 FI FI FI
IF (p2p1&2)=2 THEN status1=3
score1==+10 Position(14,1)
PrintC0(6,score1)
```



```

vpos1=75 hpos1=hpos2+65 FI
IF (p3p1&2)=2 THEN status1=9
hpos1=hpos3+5 vpos1=vpos3 FI
RETURN

PROC Note0(BYTE pitch,ctrl)
audf1=pitch audc1=ctrl aflg0=1
auf0=pitch auc0=ctrl
RETURN

PROC Note1(BYTE pitch,ctrl)
audf2=pitch audc2=ctrl aflg1=1
auf1=pitch auc1=ctrl
RETURN

PROC Sink0() ; quicksand for 0
IF scnt0=0 THEN
Note0(100+10*(10-scnt0),$AC)
IF Rand(10)=0 THEN scnt0=-1
FOR j=0 TO (10-scnt0)
DO man0(j)=0 OD k=0
FOR j=(10-scnt0) TO 10
DO man0(j)=manbou(k) k=+1 OD FI
ELSE status0=8 FI
RETURN

PROC Sink1() ; quicksand for 1
IF scnt1=0 THEN
Note1(75+10*(10-scnt1),$AC)
IF Rand(10)=0 THEN scnt1=-1
FOR j=0 TO (10-scnt1)
DO man1(j)=0 OD k=0
FOR j=(10-scnt1) TO 10
DO man1(j)=manbou(k) k=+1 OD FI
ELSE status1=8 audc2=$A0 FI
RETURN

PROC Fight0() ; bird catches 0
IF Rand(5)=0 THEN ficnt0=-1
IF ficnt0 MOD 2=0 THEN source=bird1
ELSE source=bird2 FI byt=30+Rand(50)
dest=pmbase+1792+vpos3 Note0(byt,$2C)
MoveBlock(dest,source,6)
IF ficnt0=0 THEN status0=4
FOR j=0 TO 10 DO man0(j)=carry(j) OD
FI FI
RETURN

PROC Fight1() ; bird catches 1
IF Rand(5)=0 THEN ficnt1=-1
IF ficnt1 MOD 2=0 THEN source=bird1
ELSE source=bird2 FI byt=30+Rand(50)
dest=pmbase+1792+vpos3 Note1(byt,$2C)
MoveBlock(dest,source,6)
IF ficnt1=0 THEN status1=4

```

```

FOR j=0 TO 10 DO man1(j)=carry(j) OD
FI FI
RETURN

PROC Bounce() ; Main routine
byt=Peek(106) pmpage=byt-8 Poke(106,byt-8)
DO
Graphics(1) Poke(710,244) Poke(712,150)
Poke(752,1) Poke(656,3) Poke(657,1)
Poke(708,30)
FOR j=0 TO 37 DO Put(222) OD
FOR j=3 TO 19 DO Position(1,j)
Put0(6,104) OD
Pminit() Init()
DO
hposp0=hpos0 hposp1=hpos1
hposp2=hpos2 hposp3=hpos3
DO UNTIL vcount=128 OD
SetBlock(pmbase+1024+vpos0,11,0)
SetBlock(pmbase+1280+vpos1,11,0)
; status 0 = walk player
IF status0=0 AND Rand(3)=0 THEN
hpos0=+1 IF flip0=0 THEN
Note0(120,$AC) FOR j=1 TO 10 DO
man0(j)=walk1(j) OD flip0=1 ELSE
Note0(150,$AC) FOR j=0 TO 10
DO man0(j)=walk2(j) OD flip0=0 FI
IF hpos0=120 THEN
FOR j=0 TO 10 DO man0(j)=manbou(j)
OD status0=1 FI FI
IF status1=0 AND Rand(2)=0 THEN
hpos1=+1 IF flip1=0 THEN
Note1(95,$AC) FOR j=1 TO 10 DO
man1(j)=walk1(j) OD flip1=1 ELSE
Note1(125,$AC) FOR j=0 TO 10
DO man1(j)=walk2(j) OD flip1=0 FI
IF hpos1=160 THEN
FOR j=0 TO 10 DO man1(j)=manbou(j)
OD status1=1 FI FI
; status 1 = wait to jump
IF status0=1 AND Strig(0)=0 THEN
status0=2 FI
IF status1=1 AND Strig(1)=0 THEN
status1=2 FI
; status 2 = bounce
IF status0=2 THEN Parab0() FI
IF status1=2 THEN Parab1() FI
; status 3 = cloud
IF status0=3 THEN hpos0=hpos2
IF hpos0=56 THEN status0=6 FI FI
IF status1=3 THEN hpos1=hpos2+5
IF hpos1=56 THEN status1=6 FI FI
; status 4 = bird carry
IF status0=4 THEN hpos0=hpos3+5
vpos0=vpos3+5

```

```

IF hpos0=130 THEN FOR j=0 TO 10
DO man0(j)=falling(j) OD
status0=7 audc1=$AC FI FI
IF status1=4 THEN hpos1=hpos3
vpos1=vpos3+5
IF hpos1=130 THEN FOR j=0 TO 10
DO man1(j)=falling(j) OD
status1=7 audc2=$AC FI FI
; status 5 = quicksand
IF status0=5 THEN Sink0() FI
IF status1=5 THEN Sink1() FI
; status 6 = climb down ladder
IF status0=6 THEN vpos0=+1
IF clk>10 THEN clk=0 IF flip0=0
THEN Note0(100,$AC) FOR j=1 TO 10
DO man0(j)=climb1(j) OD flip0=1
ELSE Note0(50,$AC) FOR j=0 TO 10 DO
man0(j)=climb2(j) OD flip0=0 FI FI
IF vpos0=180 THEN
Begin0() status0=0 FI FI
IF status1=6 THEN vpos1=+1
IF clk>10 THEN clk=0 IF flip1=0
THEN Note1(95,$AC) FOR j=1 TO 10
DO man1(j)=climb1(j) OD flip1=1
ELSE Note1(30,$AC) FOR j=0 TO 10 DO
man1(j)=climb2(j) OD flip1=0 FI FI
IF vpos1=175 THEN
Begin1() status1=0 FI FI
; status 7 = fall
IF status0=7 THEN vpos0=+1
audf1=vpos0
IF vpos0=botlim THEN status0=5 FI FI
IF status1=7 THEN vpos1=+1
audf2=vpos1
IF vpos1=botlim THEN status1=5 FI FI
; status 9 = caught by bird
IF status0=9 THEN fight0() FI
IF status1=9 THEN fight1() FI
source=man0 dest=pmbase+1024+vpos0
MoveBlock(dest,source,11)
source=man1 dest=pmbase+1280+vpos1
MoveBlock(dest,source,11)
stk0=Stick(0) stkl=Stick(1)
IF (Stk0&4)=0 AND hpos0m>ltlim
THEN hpos0m=-1 FI
IF (Stk0&8)=0 AND hpos0m>rtlim
THEN hpos0m=+1 FI
IF (Stk1&4)=0 AND hpos1m>ltlim
THEN hpos1m=-1 FI
IF (Stk1&8)=0 AND hpos1m>rtlim
THEN hpos1m=+1 FI
hposm0=hpos0m-4 hposm2=hpos1m-4
hposm1=hpos0m+4 hposm3=hpos1m+4
IF Peek(20) MOD 3=0 THEN hpos2=-1 FI
IF status0<9 AND status1<9 THEN

```



```

vpos3==+vdir3
IF vpos3>155 THEN vdir3=-1 FI
IF vpos3<105 THEN vdir3=1 FI
IF birdir=1 THEN source=bird1 ELSE
    source=bird2 FI
dest=pmbase+1792+vpos3
MoveBlock(dest,source,8)
hpos3==+birdir FI
IF Rand(100)=0 THEN birdir=1 FI
IF Rand(100)=0 THEN birdir=-1 FI
; status 8 = dead
IF status=8 AND status=8 THEN
    Begin0() Begin1() FI
IF aflg0=1 THEN auc0=-1 auc1=auc0
    IF (auc0&$0F)=0 THEN aflg0=0 FI FI
IF aflg1=1 THEN auc1=-1 auc2=auc1
    IF (auc1&$0F)=0 THEN aflg1=0 FI FI
IF consol=3 THEN Setdif0()
    DO UNTIL consol<>3 OD FI
IF consol=5 THEN Setdif1()
    DO UNTIL consol<>5 OD FI
UNTIL consol=6
OD OD
RETURN

```

```

a=InputB()
Close(2)
Open(2,str,8,0)
RETURN

```

;codeconvert changes Tinytext end of line  
;characters to RETURNS

```

PROC codeconvert()
FOR a=1 to str(0)
DO
IF str(a)=19 OR str(a)=5 THEN str(a)=155
FI
OD
RETURN

```

```

PROC converter()
CARD num,loop
DO
count=0
mem=$4000
Graphics(0)
PrintE("TinyText to TextWizard File Converter")
PrintE(" ")
PrintE("      Written by Dale Lutz")
PrintE(" ")
PrintE("      In ACTION!")
PrintE(" ")
inputname()
num=InputC(1)
FOR loop=1 TO num+1
DO
linein()
codeconvert()
out()

```

```

OD
outputname()
allout()
Close(1)
Close(2)
PrintE(" ")
PrintE("Conversion Complete!")
PrintE(" ")
PrintE("Hit RETURN to do another conversion")
a=InputB()

```

```

OD
RETURN
; **** LISTER ****
; by Dale Lutz

```

```

10 REM ***FUN WITH ART PICTURE LOADER*
**
15 REM ***REV.12/83 FOR F.A.C.S. MEMSL
ETTER**
20 REM ***BY HARRY PERKINS***
25 SE.2,0,0:PRINT"K":605.29000
30 PRINT"ENTER PICTURE FILE TO LOAD":I
NPUT PIC$
35 GOS.29100
40 IF PEEK(764)<>28 THEN G.40
50 GOS.29400
60 SE.2,0,0
70 G.20
29000 REM --INT FOR A FUN WITH ART--
29010 MX=7:DIM CIO$(MX)
29015 FOR I=1 TO MX:READ J:CIO$(I)=CHR
$(J):NEXT I
29020 D.104,169,16,170,76,86,228
29022 MX=15:DIM PIC$(MX)
29025 DIM DLION$(MX)
29030 FOR I=1 TO MX:READ J:DLION$(I)=C
HR$(J):NEXT I
29035 D.104,169,192,141,232,6,162,6,16
0,221
29040 D.169,6,76,92,228
29045 MX=18:DIM DLIOFF$(MX)
29050 FOR I=1 TO MX:READ J:DLIOFF$(I)=
CHR$(J):NEXT I
29055 D.104,169,64,141,232,6,141,14,21
2,162
29060 D.228,160,95,169,6,76,92,228
29070 REM INIT. VARIABLES AND STEAL 50
ME MEMORY FROM BASIC
29075 REM
29080 IOCB=848:OLDSCB=PEEK(560):OLDSCN
=PEEK(561)
29085 PICBAS=(INT(PEEK(742)/16)-2)*16
29090 DLIBAS=PICBAS-9:POKE 741,0:POKE
742,DLIBAS
29095 RET.
29100 REM -- LOAD AND SHOW A FUN WITH
ART PICTURE--
29105 REM ROUTINE RETURNS A=-1 IF SOME
THING IS WRONG WITH FILE

```



```

100 REM *****
110 REM *
120 REM * LABEL MAKER *
130 REM *
140 REM * by *
150 REM *
160 REM * Larry L Farmer *
170 REM * 387 Fiddlers Green *
180 REM * Dover, DE 19901 *
190 REM *
200 REM *****
210 REM
220 REM
225 TRAP 230:J=J+1:TRAP 40000:REM **
ADDED FOR BASIC XL
230 SETCOLOR 2,13,4:SETCOLOR 4,13,4:PO
KE 752,1
240 PRINT CHR$(125):PRINT "Insure that
printer is on line and topof form is
set."
250 PRINT:PRINT "Press RETURN to co
ntinue.":POKE 764,255
260 IF PEEK(764)=255 THEN 260
270 GOSUB 800:REM ** INITIALIZATION
280 GOTO 930:REM ** MENU
290 REM ** CREATE LABEL
300 GOSUB 880:GOSUB 1690:REM ** REINIT
IALIZE & DRAW BORDER
310 GOSUB 1090:REM ** PRINT LABEL ON S
CREEN
320 IF PAR(8)=1 THEN GOSUB 570:REM **
CENTER LINES HORIZONTALLY
330 GOSUB 690:REM ** SET VERTICAL ADVA
NCE
340 PRINT "START" for single copy; [E
LECT] for"
350 PRINT:PRINT "Multiples; [START] [
SELECT] for new"
360 PRINT:PRINT "label; [OPTION] for
menu"
370 POKE 53279,15
380 IF PEEK(53279)=15 THEN 380
390 FOR DELAY=1 TO 25:NEXT DELAY
400 CH=PEEK(53279):IF CH=6 THEN 450:RE
M ** SINGLE
410 IF CH=5 THEN 470:REM ** MULTIPLE
420 IF CH=4 THEN 300:REM ** NEW LABEL
430 IF CH=3 THEN 930:REM ** MENU
440 GOTO 370
450 N=1:GOSUB 500:GOTO 370
460 N=101:GOTO 480:REM ** ERROR TRAP
470 PRINT:PRINT "How many copies (1 t
o 100)":POKE 752,0:TRAP 460:INPUT N
480 IF N>100 THEN POSITION 2,19:PRINT

```

```

":POSITION 2,18:GOTO 470
490 GOSUB 500:GOSUB 2150:TRAP 240:GOTO
340:REM ** PRINT LABELS & ERASE INSTR
UCTIONS
499 REM ** PRINT LABELS
500 FOR I=1 TO N
510 PRINT #1;CHR$(20);REM ** VERTICAL
ADVANCE
520 FOR J=0 TO NUMLNS-1:Y=J*NUMCHRS
530 PRINT #1;TEXT$(Y+1,NUMCHRS+Y)
540 NEXT J
550 PRINT #1;CHR$(12);REM ** FORMFEED
560 NEXT I:RETURN
569 REM ** LINE CENTERING ROUTINE
570 POSITION 5,15:PRINT "P L E A S E
S T A N D B Y":POSITION 2,13:N=0
580 FOR I=0 TO NUMLNS-1
590 FOR J=NUMCHRS TO 1 STEP -1
600 Y=J+I*NUMCHRS
610 IF TEXT$(Y,V)="" THEN X=X+1:NEXT
J
620 IF J=0 OR X<2 THEN 670
630 TEMP$=TEXT$(Y-J+1,Y)
640 X=INT(X/2)
650 FOR K=1 TO X:TEXT$(Y-J+K,Y-J+K)="
":NEXT K
660 TEXT$(Y-J+K)=TEMP$
670 X=0:TEXT$(25,1)="":NEXT I
680 RETURN
689 REM ** SET VERTICAL ADVANCE
690 IF PAR(10)=1 THEN 720
700 IF PAR(5)=1 THEN ON NUMLNS GOTO 75
0,740,730,720
710 ON NUMLNS GOTO 790,700,770,730,760
720 PRINT #1;CHR$(27);"C,0,$":RETURN
730 PRINT #1;CHR$(27);"C,4,$":RETURN
740 PRINT #1;CHR$(27);"C,8,$":RETURN
750 PRINT #1;CHR$(27);"C,12,$":RETURN
760 PRINT #1;CHR$(27);"C,1,$":RETURN
770 PRINT #1;CHR$(27);"C,7,$":RETURN
780 PRINT #1;CHR$(27);"C,10,$":RETURN
790 PRINT #1;CHR$(27);"C,13,$":RETURN
799 REM ** INITIALIZATION
800 DIM PAR(10),TEXT$(25,1),TEMP$(50)
810 PRNTOEN=29
820 FOR I=1 TO 10:IF I=2 OR I=5 OR I=7
OR I=9 THEN PAR(I)=1:NEXT I
830 PAR(1)=0:NEXT I
840 OPEN #1,0,0,"P:"
850 PRINT #1;CHR$(5);CHR$(6);REM ** D
ESELECT TEXT JUSTIFY & SELECT FIXED SP
ACING

```

```

860 PRINT #1;CHR$(27);"L,40,40,$":REM
** SET FORM SIZE
870 PRINT #1;CHR$(27);"J,0,361,$":REM
** SET MARGINS
880 IF PAR(1)=1 THEN PRINT #1;CHR$(2
);REM ** SET NORMAL PRINTER MODE
890 PRINT #1;CHR$(PRNTOEN);REM ** SET
PRINT DENSITY
900 REM ** SET LINE DENSITY
910 IF PAR(5)=1 THEN PRINT #1;CHR$(27
);"B,0,$":NUMLNS=4:RETURN
920 PRINT #1;CHR$(27);"B,6,$":NUMLNS=
5:RETURN
929 REM ** MENU
930 PRINT CHR$(125):POKE 752,1:REM **
CLEAR SCREEN, TURN OFF CURSOR
940 POSITION 14,2:PRINT "MENU"
950 PRINT:PRINT "PRINT LABELS"
960 PRINT:PRINT "SET PARAMETERS"
970 PRINT:PRINT "QUIT"
980 PRINT:PRINT "Command?":POK
E 764,255
990 IF PEEK(764)=255 THEN 990
1000 CH=PEEK(764):POKE 764,255
1010 IF CH=10 THEN 300:REM ** PRINT
1020 IF CH=62 THEN 1050:REM ** PARAM
ETERS
1030 IF CH=47 THEN CLOSE #1:PRINT:PRI
NT "Have a nice day!":POKE 752,0:END:
REM ** QUIT
1040 GOTO 930
1049 REM ** PARAMETER MENU
1050 PRINT CHR$(125):POSITION 14,0:PRI
NT "PARAMETERS"
1060 POSITION 2,2:PRINT "PRINT SIZE:"
1070 POSITION 4,4:PRINT "(1) ENHANCED"
1080 POSITION 4,5:PRINT "(2) LARGE"
1090 POSITION 4,6:PRINT "(3) MEDIUM"
1100 POSITION 4,7:PRINT "(4) SMALL"
1110 POSITION 26,2:PRINT "LINES/LABEL:
"
1120 POSITION 29,4:PRINT "(5) 4"
1130 POSITION 29,5:PRINT "(6) 5"
1140 POSITION 2,9:PRINT "LINE POSITION
5:"
1150 POSITION 4,11:PRINT "(7) LEFT JU
STIFIED"
1160 POSITION 4,12:PRINT "(8) CENTERED
"
1170 POSITION 26,8:PRINT "VERTICAL":PO
SITION 26,9:PRINT "CENTERING?"
1180 POSITION 30,11:PRINT "(9) YES"
1190 POSITION 29,12:PRINT "(10) NO"
1200 POSITION 11,15:PRINT "(11) PRINT
LABELS"

```

A-F SALADS  
G-K CASSEROLES  
L-P DESSERTS  
Q-Z POTPOURRI (whatever)

## FOR POTLUCK PICNIC



LABEL MAKER con't.

```

1210 PRINT "_____
"
1220 FOR I=1 TO 4
1230 ON I GOSUB 1490,1540,1590,1640
1240 NEXT I
1250 POSITION 2,10:PRINT "Number of pa
parameter you want set";POKE 752,0
1259 REM ** RESET PARAMETERS
1260 TRAP 1320:INPUT N:N=INT(N)
1270 IF N)=1 AND M<=4 THEN GOSUB 1340:
REM ** PRINT DENSITY
1280 IF N=5 OR N=6 THEN GOSUB 1400:REM
** LINE DENSITY
1290 IF N=7 OR N=8 THEN GOSUB 1430:REM
** LINE POSITION
1300 IF N=9 OR N=10 THEN GOSUB 1460:RE
M ** VERTICAL CENTERING
1310 IF N=11 THEN TRAP 240:GOTO 300
1320 POSITION 35,10:PRINT " "
1330 POSITION 34,10:GOTO 1260
1340 FOR I=1 TO 4:PAR(I)=0:NEXT I:PAR(
N)=1:GOSUB 1490
1350 RESTORE 1350+10*M:READ PRNTDEN:RE
TURN
1360 DATA 1
1370 DATA 29
1380 DATA 30
1390 DATA 31
1400 IF N=5 THEN PAR(5)=1:PAR(6)=0:GOT
O 1420
1410 PAR(5)=0:PAR(6)=1
1420 GOSUB 1540:RETURN
1430 IF N=7 THEN PAR(7)=1:PAR(8)=0:GOT
O 1450
1440 PAR(7)=0:PAR(8)=1
1450 GOSUB 1590:RETURN
1460 IF N=9 THEN PAR(9)=1:PAR(10)=0:GO
TO 1480
1470 PAR(9)=0:PAR(10)=1
1480 GOSUB 1640:RETURN
1489 REM ** PRINT ASTERISKS BY ACTIVE
PARAMETERS
1490 FOR J=1 TO 4
1500 POSITION 3,J+4:IF PAR(J)=1 THEN P
RINT "M":GOTO 1520
1510 PRINT " "
1520 NEXT J
1530 RETURN
1540 FOR J=5 TO 6
1550 POSITION 20,J-1:IF PAR(J)=1 THEN
PRINT "M":GOTO 1570
1560 PRINT " "
1570 NEXT J
1580 RETURN
1590 FOR J=7 TO 8
1600 POSITION 3,J+4:IF PAR(J)=1 THEN P
RINT "M":GOTO 1620
1610 PRINT " "
1620 NEXT J
1630 RETURN
1640 FOR J=9 TO 10
1650 POSITION 20,J+2:IF PAR(J)=1 THEN
PRINT "M":GOTO 1670
1660 PRINT " "
1670 NEXT J
1680 RETURN
1689 REM ** DRAW BORDER & PRINT INSTRU
CTIONS
1690 PRINT CHR$(125)
1700 FOR I=1 TO 4:IF PAR(I)=1 THEN 172
0
1710 NEXT I
1720 RESTORE 1840+10*I:READ NUMCHRS
1730 IF PAR(4)=1 THEN NUMLNS=2*NUMLNS
1740 POSITION 2,0
1750 FOR I=1 TO NUMCHRS:PRINT CHR$(14)
;:NEXT I
1760 FOR I=1 TO NUMLNS:POSITION NUMCHR
S+2,I:PRINT CHR$(22):NEXT I
1770 FOR I=1 TO NUMCHRS:PRINT CHR$(13)
;:NEXT I
1780 FOR I=NUMLNS TO 1 STEP -1:POSITIO
N 1,I:PRINT CHR$(2):NEXT I
1790 POSITION 2,13:PRINT "To End type
CTRL M":POSITION 5,14:PRINT CHR$(1
3)
1800 PRINT "To Restart type CTRL R
":POSITION 5,16:PRINT CHR$(13)
1810 PRINT "For MENU type CTRL M":
POSITION 6,18:PRINT CHR$(13)
1820 POSITION 3,1:POKE 752,0:PRINT CHR
$(30);
1830 IF PAR(4)=1 THEN NUMLNS=NUMLNS/2:
NUMCHRS=2*NUMCHRS
1840 RETURN
1850 DATA 15
1860 DATA 30
1870 DATA 36
1880 DATA 25
1889 REM ** CREATE LABEL ON SCREEN
1890 OPEN #2,4,0,"K":CPOS=1
1900 TEXT$=" ":TEXT$(250)=TEXT$(1):TEX
T$(2)=TEXT$(1):REM ** FILL WITH BLANKS
1910 FOR I=1 TO NUMLNS
1920 FOR J=1 TO NUMCHRS
1930 IF PAR(4)=1 AND J=26 THEN PRINT
1940 GET #2,K5
1950 IF K5)31 AND K5<123 THEN 2000
1960 IF K5<126 THEN 2000:REM ** BACKS
PACE
1970 IF J=1 THEN 1940
1980 IF PAR(4)=1 AND J=26 THEN POSITIO
N 27,PEEK(84)-1:PRINT CHR$(K5);
1990 CPOS=CPOS-1:J=J-1:TEXT$(CPOS,CPOS
)=" ":PRINT CHR$(K5);:GOTO 1940
2000 IF K5<155 THEN 2030:REM ** EOL
2010 IF I<NUMLNS THEN CPOS=INT((CPOS-1
)/NUMCHRS)*NUMCHRS+NUMCHRS+1:GOTO 2100
2020 GOTO 2110
2030 IF K5=5 THEN NUMLNS=I:GOTO 2120:R
EM ** END
2040 IF K5=18 THEN CLOSE #2:GOSUB 2210
:GOTO 1890:REM ** RESTART
2050 IF K5=13 THEN CLOSE #2:POP :GOTO
930:REM ** MENU
2060 IF K5)27 AND K5<32 THEN 1940:REM
** IGNORE CURSOR CONTROL COMMANDS
2070 IF K5=156 OR K5=157 OR K5=254 OR
K5=255 THEN 1940:REM ** IGNORE INSERT
& DELETE COMMANDS
2080 TEXT$(CPOS,CPOS)=CHR$(K5):CPOS=CP
OS+1:PRINT CHR$(K5);
2090 NEXT J
2100 PRINT :IF PAR(4)=1 AND J<=25 THEN
PRINT
2110 NEXT I
2120 IF J=1 THEN NUMLNS=I-1
2130 CLOSE #2
2140 REM ** ERASE INSTRUCTIONS
2150 POSITION 2,13:POKE 752,1
2160 FOR I=1 TO 7
2170 PRINT " "
NEXT I
2180 POSITION 2,13
2190 RETURN
2200 REM ** ERASE LABEL
2210 IF PAR(4)=1 THEN NUMLNS=2*NUMLNS:
NUMCHRS=NUMCHRS/2
2220 FOR I=1 TO NUMLNS:POSITION 2,I
2230 FOR J=1 TO NUMCHRS:PRINT " ";:NEX
T J
2240 NEXT I:POSITION 3,1:PRINT CHR$(30
);
2250 IF PAR(4)=1 THEN NUMLNS=NUMLNS/2:
NUMCHRS=2*NUMCHRS
2260 RETURN

```



NO MEETING THIS MONTH  
AUGUST MEETING IN JASPER PARK  
7:00PM, AUGUST 15TH



## RUTH'S PILOT

Winding down, this is the last of the series of articles about using strings in PILOT. The last technique to be covered in this series is the ability to choose parts of a string. The program at the end of this article demonstrates such routines, some of the string principles we have been discussing, and uses one of the alphabets available on the new PILOT disk.

Building upon the MS: techniques in last month's article, one can easily select parts of strings to be used in programs. Line 50 creates \$ALPHA made up of the letters of the alphabet separated by commas. As explained, the commas are separators and become \$MATCH in the program. In this particular program random number generation is used to select a letter of alphabet. In other programs the number could be selected by the programmer or user in this type of routine. A counter is used to match to the first comma as many times desired. The routine matches the first comma it comes to in the string, makes the comma \$MATCH, places the letter before the comma in \$LEFT, and places the remainder of \$ALPHA in \$RIGHT. \$RIGHT must be placed in the accept buffer to be reused each time through the routine until the counter reaches 0.

\$LEFT is placed in the accept buffer to designate the letter displayed on the graphics screen. \$LEFT passes its value to \$LETTER in line 210 so it can be used with the match command to determine right and wrong choices.

We have used the smiley face as the reinforcement in this program because it is our preschooler's favorite. A little more fun can be added by making the face wink, etc. by replacing the characters shown and using the POS. command.

This alphabet is our favorite lower case alphabet because of its size. The J:\*MATCH statement at the end of each letter will have to be removed to use it elsewhere (they are not on the PILOT disk). Using string techniques makes it possible to do many things with alphabets and word type programs. This simple program is just one example of the kind of thing which can be done. "Unwind" a little, it can be challenging, and sometimes surprising what one finds at the end of a string.

— Ruth Ellsworth

## AGENT U.S.A.

AGENT U.S.A. by Scholastic Wizware is, without a doubt, the favorite game software of our 11 and 13 year olds, and assorted friends. The loss of interest usually setting in after the children have played a game a few times has been totally absent on this one. Fortunately the sound accompanying this game is not displeasing as it has become almost background music in our home over the last month and a half.

This piece of software is a cleverly disguised geography and map lesson. The object of the game is to find and defuse the "FuzzBomb" which turns everyone it touches into "FuzzBodies." At the beginning of the game the player is placed in a randomly chosen city train station. All the cities in the game are linked by train or in some cases by monorail. The player becomes "Agent U.S.A." and uses the train system to travel from city to city in order to find and destroy the "FuzzBomb." Maps and information concerning the "Fuzz Menace" are located in the Information Booth at the train station of all state capitals. Our children have spent hours studying the Atlas and planning their trips through this game as "Agent U.S.A."

We highly recommend this game. We wish there were more like it. It is educational, challenging, and fun. The children insist it is never the same from game to game and they love to play it "just to see what will happen."

—Ruth Ellsworth

## SPACE KNIGHTS

SPACE KNIGHTS by Reston Publishing Company is an interesting innovation in adventure software. This package comes with a book of science fiction adventures and disk of programs to put the reader into the action of the story.

This package is well worth the price. Nine games are included on the double sided disk. Of these our favorites were: GAMMALON ENCOUNTER, a code cracking then "shootem up" game; WAR ROOM, a dungeon type game; BUG BUSTER, a "find and destroy the enemy" game; and WEOMBY, a good landing simulation.

The games in general were not easy, in fact, the MYSTERY GAME is really that — a real head bender.

Our problem with this package is it comes on a speed sensitive disk we have not yet been able to get our drive to read, so we have had to go visiting to use this software (no small trick with a house full of kids). I must say, however, that Reston bent over backwards trying to provide a disk which would work with our system.

If you are interested in science fiction and want to experience some of the "feeling of being there," or if you like the variety of not so easy games, this package will probably meet all your expectations. It requires 24K minimum, a BASIC cartridge, joysticks and paddles.

— Ruth Ellsworth

## MAIL-MERGE

(Reprint: May, 1984 Redwood Atari Group)

AtariWriter provides for creating form letters by using [OPTION][INSERT]. This function halts the printing and prompts the user for input at each point in the letter which needs to be individualized. You must hang around the printer, so it might not be practical for you.

An interoffice memo at Atari, Inc. reveals a way around this limitation. The [OPTION][INSERT] feature can only handle 35 characters at a time. The manual warns you to make a list if you have several "blanks" to be filled in, because you cannot see the text as it is printing. The key to the mail-merge feature is to create a data file (the list) which contains each "fill-in" in sequential order. Then you chain the file while printing.

Here's a sample letter using [O] for [OPTION][INSERT] and [R] for [RETURN]: [O][O]  
Dear[O],

It sure was great to get your letter of last [O]! I've been [O]. It sounds like you've been [O]. Let's keep in touch!  
[O]

Here's a sample data file:

[R][R]Mom[R]month[R]well[R]well, too[R]Love,[R]

Here's another sample data file:

[R][R]Buford[R]year[R]busy, I'm still busy, and I can't talk now[R]busy yourself, after graduating from Folsom. . .[R]Sincerely,[R]

To use the mail-merge, create your form letter using [OPTION][INSERT]. Then create your data file in sequence, making sure each item is followed by [RETURN]. You must be sure to have the same number of items in your data file as are to be entered in your letter, otherwise the subsequent letters could be messed up. The [RETURN] acts as the delimiter for the data — AtariWriter goes back to the original and continues printing when [RETURN] is encountered until the next [OPTION][INSERT] occurs.

Print the letter. When the prompt reads "MAKE ENTRY, PRESS RETURN," you enter [CONTROL][V], the chaining command, and D:DATAFILENAME in capital letters. Printing will resume and AtariWriter won't bother you again until it is out of data (or done).

— Mary Varley

## SALVAGE 410

(Reprint: June, 1984, Suburban Chicago Atarians)

The electronics of the Atari 410 recorder can be used with a stereo cassette recorder for data storage. The 410 circuit itself can be used as an interface. Maybe your 410 has its head out of alignment and you cannot get the unit to work dependably. Maybe the motor speed is off or does not run at all. This "how-to" article will allow you to salvage your 410 as long as the electronics are ok. Since it's solid state, I guess the electronic part is more reliable than the mechanical part.

Figure 1 shows the 410 printed circuit board with a partial path layout for easy identification. The black, filled-in areas signify the copper paths. The white dots within the black areas are soldered connections. All you have to do is run a few jumper wires from the circuit board to a couple of jacks mounted in the 410 case. First, locate, drill and mount two jacks into the bottom half of the 410 plastic housing. I used phono jacks (Radio Shack #274-346) to keep the connections standard between the 410 and my stereo cassette deck. Mount the jacks on the back, or on the side near the back. There is plenty of room there. Next, carefully solder the jumper wires to the printed circuit board. I used shielded cable because I had some. You can probably use non-shielded wire if you keep the length short. One wire connects one jack to the point on the board labelled "Input to Filter Section". The jack ground is wired to the "Ground Path". The brown wire from the I/O cable is soldered to the board to the other jack. You can wire the two jack ground connections together. Be careful when soldering to the printed circuit board. You could add a coupling capacitor between each jack and the other lead to the jumper wire. Use 0.1 mfd capacitors (Radio Shack #272-1069 or equivalent). Be sure to keep the wires properly insulated.

The jack with the brown wire going to it should be connected to the right channel input of your stereo recorder. This wire carries the signal out of your computer. Set your recording level to 0db as you do normally. The other jack is connected to your stereo recorder's right channel output. This signal is injected into the 410's filter circuit which then goes to the computer. The signal level from your stereo recorder is not critical.

The stereo recorder has to be manually started and stopped. When you press the RETURN key on the computer, you have to start the stereo recorder in either the play or the record mode. If you play back a multi-load tape, you are going to have to stop and start the stereo recorder between the appropriate sections.

Hopefully, this setup will help someone use Atari program tapes with their mechanically defective 410. It may not be as convenient to use as a properly working 410, but for the price of two jacks, a little wire, and a little time, you can get your cassette-based system up and working again.

— Mike Chesta



## DISK N' DATA

by Arnie Silverstein

### "BOOT FILE LOADER"

Reviewing our earlier article, (BB vol I,#2) boot files are sequential files, loaded one sector after another in numerical order until done. First, sector one, bytes zero and one on DOS disks say that three sectors are to be read. Then, bytes two and three indicate the load address. Byte six is the first actual instruction to be executed. In a boot disk, this information can load the entire program.

This month's program is a boot loader that allows some binary files to be loaded without DOS being present on the disk. The program writes a menu program into the boot sectors in the place where DOS normally writes as a disk is formatted. Using a new disk, the program will first format the disk (if desired) and then it will overwrite the boot sectors. To use the program, use the "MENU MAKER" option from the main menu of this disk. The program is contained on this disk in the file "BOOTWRIT.BAS".

Once this is done, use the DOS "Q" function (Duplicate File) to get your binary programs onto the disk. On booting the disk, test the menu on your particular program to see if it will run. Not all will, but enough will for you to come to love this short fast loader for games and utilities that don't require DOS.

Please note that DOS will no longer function on any disk altered by this program. You are limited to ten programs per disk side.

Source code for the disk writing portion of the program is supplied on side 2 of this disk as "DISKWRIT.ASM". The routines used here are relocatable and take the form of a machine language string. The data to be written into the boot sectors is contained in a "DATA string". The Resident Disk Handler is used by the program.

Reader comments, problems and suggestions are requested so that this loader might be improved.

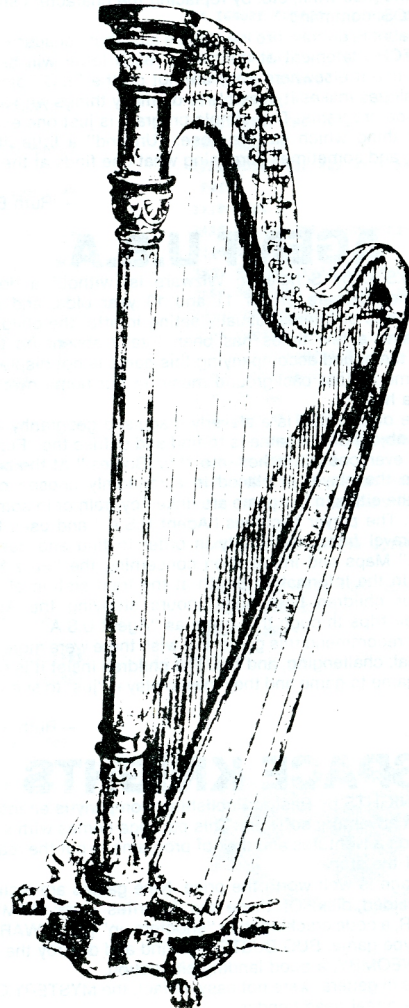
## TYPING 1

TYPING 1 (Emerald Valley Micro Consultants, P.O. Box 2605, Eugene, Or., 97402, \$30) is a true typing program for the Atari and not a game which also teaches typing as you play it. This program starts with the basics of how you should sit, arms a certain way, eyes a certain way, and where your hands and fingers should be. From that you are given certain typing exercises to perform and then you can find out your speed etc. As you practice and go from lesson to lesson you will find you have improved your typing if you knew how to type and if you didn't you will find you are learning to type.

The manual is designed like some of the typing manuals used in school, and it stands up on its own. The pages flip over as you finish with them. There are many diagrams to show you what keys you are using and where your hands should be, and at the same time the text explains what should be occurring with the lesson you are doing.

I found by following the manual and doing the lessons my typing was improved in just a short period of time using this program. It is a program for the beginner and the experienced typist as well.

—Larry Gold





# GENISIS

(Datasoft \$30)

The spiders gnaw away at the tunnel, as you blast them with your venom. The spiders will eat away the sides of the pit, limiting your movement to the walls that haven't already crumbled under the spiders' jaws.

To add to the frenzy is a purple brick sliding along the rim of the pit. Occasionally, this brick is turned to green. Touching a purple brick is certain death, but with a green brick in hand, you may repair the tunnel, getting double the points in the meantime.

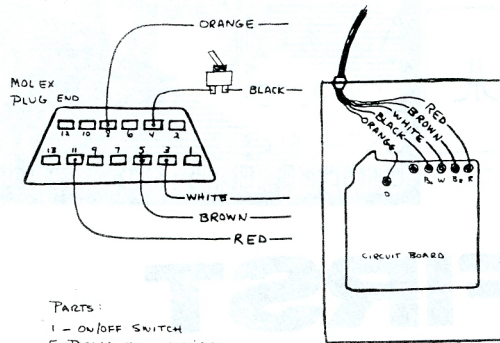
After a certain number of spiders have been eliminated, you advance to the next, and more difficult wave. This game is equipped with the usual arcade options: a pause feature, one or two player options, and a difficulty selection.

Though it is an enjoyable game, it doesn't quite have enough depth or purpose to it. I want to describe what happened the first time my little brother, the arcade game fanatic, tried this game. After his game, he looked blankly at the screen for a few moments, then turned toward me, expecting an explanation. "Is that all?" he questioned. The graphics and sound are O.K., but could be much better.

—Tim Ebling

A hand-drawn schematic diagram of the button side of a component. The diagram shows a rectangular outline with a complex internal structure. A large, solid black area on the left represents a component, possibly a filter section. A line labeled "GROUND PATH" points to a series of dots connecting this black area to other parts of the circuit. Another line labeled "INPUT TO FILTER SECTION" points to a specific connection point. A line labeled "BROWN WIRE FROM I/O CABLE" points to a connection on the right side of the component. The text "BUTTON SIDE" is written at the bottom.

Farewell



NOTE: IN NEWER RECORDERS WIRES FROM CABLE ARE SOLDERED TO CIRCUIT BOARD AS SHOWN HERE, SIZES AND SHAPES OF CIRCUIT BOARDS WILL VARY



## Atari Computer Enthusiasts

A.C.E. is an independent, non-profit and tax exempt computer club and user's group with no connection to the Atari Company, a division of Warner Communication Company. We are a group interested in educating our members in the use of the Atari Computer and in giving the latest News, Reviews and Rumors.

All our articles, reviews and programs come from you, our members.

Our membership is world-wide; membership fees include the A.C.E. Newsletter. Dues are \$12 a year for U.S., and \$22 a year Overseas Airmail and include about 10 issues a year of the ACE Newsletter.

**Subscription Dep't:** 3662 Vine Maple Dr., Eugene, OR 97405.

\* President— Robert Browning, 90 W. Myoak Dr., Eugene, OR 97404  
503-689-1513

Vice Pres— Larry Gold, 1927 McLean Blvd., Eugene, OR 97405  
503-686-1490

Secretary— Bruce Ebling, 1501 River Loop #1, Eugene, OR 97404  
503-688-6872

**Librarian— Ron and Aaron Ness, 374 Blackfoot, Eugene, OR 97404  
503—689-7106.**

Editors—

Mike Dunn, 3662 Vine Maple Dr., Eugene, OR 97405 / 503-344-6193

Jim Bumpas, 4405 Dillard Rd., Eugene, OR 97405 503 / 484-9925

E.R.A.C.E. (Education SIG Editor)—

Ali Erickson, 295 Foxtail Dr., Eugene, OR 97405 / 503-687-1133

E.R.A.C.E. Corresponding Secretary—

Robert Browning, 90 W. Myoak, Eugene, OR 97404 / 503-689-1513

Send 50c stamps or coin (\$1 overseas) to the Ness' for the new, updated ACE Library List-new in May 84 !

### Best of ACE books

Volume 1 contains bound issues of the ACE Newsletter from the first issue, Oct 81 to June of 1982

Volume 2 covers July 1982 to June 1983

Only \$12 each (\$2 extra for Airmail). Available only from:

George Suetsugu  
45-602 Apuapu St  
Kaneohe, HI 96744

### SortFinder 1.2

A composite index of Atari related articles from 5 popular computer periodicals from Apr '81 to June '83, including ACE. Only \$6 for ACE member from:

Jim Carr, Valley Soft  
2660 S.W. DeArmond  
Corvallis, OR 97333

## TYPESETTING FROM YOUR COMPUTER

**ATARI OWNERS:** If you have a modem, text editor, and communications program to send ASCII files, you should consider the improved readability and cost savings provided by **TYPESETTING** your program documentation, manuscript, newsletter, or other lengthy text instead of just reproducing it from line printer or daisy-wheel output. Computer typesetting by telephone offers you high quality, space-saving copy that creates the professional image you want! Hundreds of type styles to choose from with 8 styles and 12 sizes "on line." And it's easy to encode your copy with the few typesetting commands you need.

**COMPLETE CONFIDENTIALITY GUARANTEED**

— Bonded for your protection —

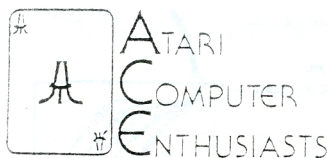
**PUBLICATION DESIGN, EDITING, & PRODUCTION**

**Editing & Design Services**

**30 East 13th Avenue Eugene, Oregon 97401**  
**Phone 503/683-2657**

### Bulletin Board (503) 343-4352

On line 24 hours a day, except for servicing and updating. Consists of a Tara equipped 48K Atari 400 with a TARA keyboard, 2 double-density double sided disk drives with an ATR 8000 interface, 2- 8" double density disk drives, an Epson MX80 printer, a Hayes SmartModem; running the ARMUDIC Bulletin Board software written by Frank L. Hubbard, 1206 N. Stafford St., Arlington, VA 22201. See the Nov '82 issue for complete details.



3662 Vine Maple Dr. Eugene OR 97405

# FIRST CLASS MAIL